Amendments to the Claims

Please amend Claims 1, 13, 17, 25, 27, 29, 30, 31, 33 and 38. Please add new Claims 43-45. The Claim Listing below will replace all prior versions of the claims in the application:

Claim Listing

1. (Currently Amended) A queue comprising:

a first memory having a first memory access time;

a second memory having a second memory access time; and

control logic which enqueues in the queue a pointer to data to be transmitted by writing the pointer in the first memory, ~~and transferring~~ transfers the pointer to the second memory and dequeues the pointer from the second memory.

2. (Original) A queue as claimed in Claim 1 wherein the first memory access time is less than the second memory access time.

3. (Original) A queue as claimed in Claim 1 wherein the control logic transfers a plurality of pointers to the second memory in a single transfer cycle.

4. (Original) A queue as claimed in Claim 1 wherein the control logic enqueues the pointer in the first memory in a single write operation and establishes a linked list of pointers in the second memory after the write operation.

5. (Original) A queue as claimed in Claim 1 wherein the control logic dequeues the pointer by reading the pointer from the second memory.

6. (Original) A queue as claimed in Claim 1 wherein the control logic fills a cache row in the first memory before transferring the cache row into the second memory.

7. (Original) A queue as claimed in Claim 6 wherein the first memory transfers the cache row in a single write operation.

8. (Original) A queue as claimed in Claim 1 wherein the control logic partially fills a cache row in the first memory before transferring the cache row into the second memory in a single write operation.

9. (Original) A queue as claimed in Claim 1 wherein entries in a cache row in the first memory are ordered by position in the cache row.

10. (Original) A queue as claimed in Claim 1 wherein the first memory includes two cache rows.

11. (Original) A queue as claimed in Claim 1 wherein a packet vector stored in the second memory includes a cache row entry and a count of the number of pointers stored in a cache row entry.

12 (Previously Presented) A queue as claimed in Claim 1 wherein a packet vector stored in the second memory includes a link to a next packet vector in the queue.

13. (Currently Amended) A queuing method comprising the steps of:

writing in a first memory having a first memory access time a pointer to data to be transmitted; ~~and~~

transferring the pointer to a second memory having a second memory access time; and

dequeuing the pointer from the second memory.

14. (Original) A queuing method as claimed in Claim 13 wherein the first memory access time is less than the second memory access time.

15. (Original) A queuing method as claimed in Claim 13 wherein the step of transferring forwards a plurality of pointers to the second memory in a single transfer cycle.

16. (Original) A queuing method as claimed in Claim 13 wherein the step of writing writes the pointer in a single write operation to the first memory and establishes a linked list of pointers after the write operation.

17. (Currently Amended) A queuing method as claimed in Claim 13 ~~further comprising~~ wherein the pointer is dequeued by reading ~~step of:~~

~~dequeuing~~ the pointer from the second memory.

18. (Original) A queuing method as claimed in Claim 13 wherein the step of transferring forwards a full cache row into the second memory.

19. (Original) A queuing method as claimed in Claim 13 wherein the step of transferring forwards a partially filled cache row into the second memory.

20. (Original) A queuing method as claimed in Claim 18 wherein the cache row is transferred in a single write cycle.

21. (Original) A queuing method as claimed in Claim 13 wherein entries in a cache row in first memory are ordered by position in the cache row.

22. (Original) A queuing method as claimed in Claim 13 wherein the first memory includes two cache rows.

23. (Original) A queuing method as claimed in Claim 13 wherein a packet vector stored in the second memory includes a cache row and a count of the number of pointers stored in the cache row.

24. (Previously Presented) A queuing method as claimed in Claim 13 wherein a packet vector stored in the second memory includes a link to a next packet vector.

25. (Currently Amended) A queue comprising:

a first memory having a first memory access time;

a second memory having a second memory access time; ~~and~~

means for ~~controlling the queue enqueues~~ enqueuing in the queue a pointer to data to be transmitted by writing the pointer in the first memory; ~~and~~

means for transferring the pointer to second memory; and

means for dequeuing the pointer from second memory.

26. (Original) A queue as claimed in Claim 25 wherein the first memory access time is less than the second memory access time.

27. (Currently Amended) A queue as claimed in Claim 25 wherein the means for ~~controlling the queue~~ transferring transfers a plurality of pointers to second memory.

28. (Original) A queue as claimed in Claim 27 wherein the plurality of pointers are transferred in a single transfer cycle.

29. (Currently Amended) A queue as claimed in Claim 27 wherein the means for ~~controlling the queue~~ enqueuing enqueues the pointer in a single write operation to the first memory

and the means for transferring establishes a linked list of pointers after the write operation.

30. (Currently Amended) A queue as claimed in Claim 27 wherein the means for ~~controlling the queue~~ dequeuing dequeues the pointer by reading the pointer from the second memory.

31. (Currently Amended) A queue as claimed in Claim 27 wherein the means for ~~controlling the queue~~ enqueuing fills a cache row in the first memory before ~~transferring~~ the cache row is transferred into the second memory.

32. (Original) A queue as claimed in Claim 31 wherein the cache row is transferred to the second memory in a single write operation.

33. (Currently Amended) A queue as claimed in Claim 25 wherein the means for ~~controlling the queue~~ enqueuing partially fills a cache row in the first memory before ~~transferring~~ the cache row is transferred into the second memory.

34. (Original) A queue as claimed in Claim 25 wherein entries in a cache row in the first memory are ordered by position in the cache row.

35. (Original) A queue as claimed in Claim 25 wherein the first memory includes two cache rows.

36. (Original) A queue as claimed in Claim 25 wherein a packet vector stored in the second memory includes a cache row entry and a count of the number of pointers stored in a cache row entry.

37. (Previously Presented) A queue as claimed in Claim 25 wherein a packet vector stored in the second memory includes a link to a next packet vector in the queue.

38. (Currently Amended) ~~A~~ An apparatus for storing a pointer list comprising:

a first memory having a first memory access time;

a second memory having a second memory access time; and

control logic which adds ~~in the pointer list~~ a pointer ~~to data to be transmitted~~ to the pointer list by writing the pointer in the first memory, ~~and transferring~~ transfers the pointer to the second memory and removes the pointer from the pointer list by reading the pointer from the second memory.

39. (Original) A queue as claimed in Claim 11 wherein a packet vector stored in the second memory includes a link to a next packet vector in the queue.

40. (Original) A queuing method as claimed in Claim 19 wherein the cache row is transferred in a single write cycle.

41. (Original) A queuing method as claimed in Claim 23 wherein a packet vector stored in the second memory includes a link to a next packet vector.

42. (Original) A queue as claimed in Claim 36 wherein a packet vector stored in the second memory includes a link to a next packet vector in the queue.

43. (New) The queue of Claim 1, wherein the first memory is a static random access memory and the second memory is a dynamic random access memory.

44. (New) The queuing method of Claim 13, wherein the first memory is a static random access memory and the second memory is a dynamic random access memory.

45. (New) The apparatus of Claim 38, wherein the first memory is a static random access memory and the second memory is a dynamic random access memory.